

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-281896

(43)公開日 平成7年(1995)10月27日

(51)Int.Cl.⁶

G 0 6 F 9/38

識別記号

3 7 0 B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 3 F D (全 15 頁)

(21)出願番号 特願平6-102275

(22)出願日 平成6年(1994)4月15日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 伊藤 元久

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 釜田 栄樹

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 笹岡 茂 (外1名)

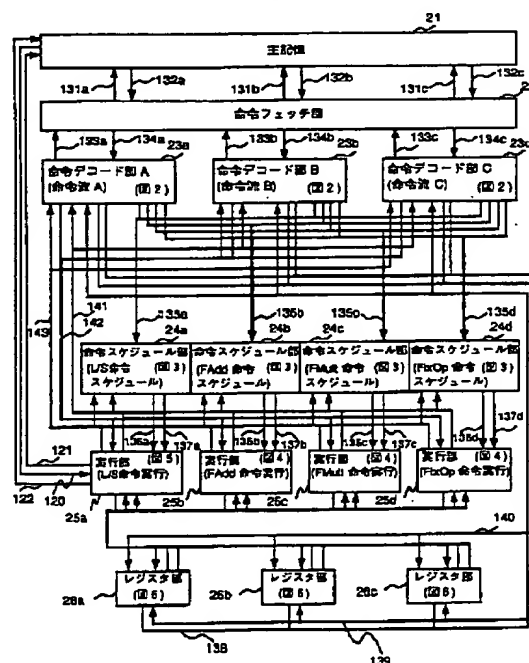
(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】 実行時に生じる実行部の空きを解消し、高効率で実行部を使用し、高い命令実行の並列性を得ることにある。

【構成】 n個の演算命令からなる長語命令形式の命令を持ち、m本の命令流を同時に実行する情報処理装置であり、実施例の場合、3本の命令流に合わせて3個のデコード部と、4個の演算命令に合わせて、演算命令対応に4個の命令スケジュール部と、演算命令対応に4個の実行部から構成される。命令デコード部は夫々独立に動作し、リソース競合関係とデータ依存関係を解明する手段と、命令発行を制御する手段からなり、前の命令が実行中であっても、リソース競合関係がなく、かつ、データ依存関係がない場合後続の命令を夫々独立に各命令スケジュール部発行する。各命令スケジュール部は、複数の命令流間に対応する実行部に送る演算命令をスケジュールリングする。

【図1】



1

【特許請求の範囲】

【請求項 1】 演算の種類毎に分類された n フィールド (n は 1 以上) の演算フィールドを有し、演算フィールド内の各演算命令が他の演算フィールド内の演算命令と互いに依存関係が無いように構成された長語命令からなる命令流 m 本 (m は 1 以上) の処理を行なう情報処理装置であって、

m 個の命令デコード部と、

該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられた n 個の命令スケジュール部と、

該 n 個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール部から発行された演算命令を実行する実行部と、

レジスタ部を備え、

前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間におけるレジスタのデータ依存関係およびリソース競合に基づき、実行中の長語命令内の演算命令全ての実行が完了していなくとも前記命令バッファ内の長語命令の前記命令スケジュール部への発行を制御する手段を備えることを特徴とする情報処理装置。

【請求項 2】 演算の種類毎に分類された n フィールド (n は 1 以上) の演算フィールドを有し、演算フィールド内の各演算命令が他の演算フィールド内の演算命令と互いに依存関係が無いように構成された長語命令からなる命令流 m 本 (m は 1 以上) の処理を行なう情報処理装置であって、

m 個の命令デコード部と、該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられた n 個の命令スケジュール部と、該 n 個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール部から発行された演算命令を実行する実行部と、レジスタ部を備え、

前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間における、レジスタのデータ依存関係を判定する判定手段とリソース競合を判定する判定手段と、該両判定手段の判定結果がレジスタのデータ依存関係がなく且つリソース競合がないことを示すとき、実行中の長語命令内の演算命令全ての実行が完了していなくとも命令バッファ内の長語命令を各演算命令に分け対応する命令スケジュール部へ発行制御する手段を備え、

前記各命令スケジュール部は、前記命令デコード部から発行された演算命令を保持する命令スケジュールバッファと、命令スケジュールバッファに保持された演算命令の中から実行部に投入する演算命令を選択し、該選択した演算命令を命令流番号と共に実行部に向け投入制御する手段を備え、

前記各実行部は、前記命令スケジュール部から投入され

2

る演算命令を実行する演算命令実行手段と、命令スケジュール部から投入される命令流番号を受け取りこれを保持する手段と、前記投入された演算命令と命令流番号から、前記演算命令実行手段の動作を制御する手段と、前記演算命令実行手段の状態を前記命令デコード部および命令スケジュール部に通知する手段を備えることを特徴とする情報処理装置。

【請求項 3】 請求項 2 記載の情報処理装置において、前記実行部の演算命令実行手段は、演算命令の実行により例外発生を検知したとき例外発生信号を前記命令流番号に対応する命令デコード部に通知し、前記デコード部は、例外発生信号を通知した実行部に対して、前記全ての実行部の演算命令実行手段の状態が受け入れ可能状態のとき、例外発生をした演算命令の再実行を指示する手段を備えることを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、長語命令方式を用いて並列実行性を高め、性能向上を計る情報処理装置に関する。

【0002】

【従来の技術】 図 7 に示すのは、長語命令方式 (以下、VLIW 方式) 計算機の一例である。VLIW 方式は、長い命令を多数の演算フィールドに分割し、各々の演算フィールドで実行部、レジスタ、メモリ等のリソースを独立して制御し、並列処理を実現する。VLIW 方式では、コンパイル時に並列性の抽出を行なう。コンパイラはプログラムの中から並列実行可能な演算命令を取りだし、1 つの長語命令に合成する。VLIW 方式では、コンパイル時にデータ依存関係を始めたとした演算命令間の依存関係のチェックを行ない、実行時には依存関係のチェックはしない。そのため、依存関係チェック用のハードウェアが不要となり、ハードウェア量を削減することができる。

【0003】 しかし、命令間の依存関係をチェックするハードウェアを持たないため、実行部の空きが生じることがある。命令間の依存関係をチェックするハードウェアを持たない VLIW 方式では、無矛盾性の保障のため、先の長語命令の終了を待ってからでないと次の長語命令を実行できない。このため、長語命令内に実行時間が他の演算フィールドの演算命令と異なる演算命令が存在すると、最も長くかかる演算命令の終了を待ってから次の長語命令の実行が開始されるため、演算命令が実行されない実行部が生じる。実行時間に差が生じる例としては、データのロード/ストア命令がある。データのロード/ストア命令は、キャッシュのヒット/ミスヒットにより実行時間が異なる。また、実行部の空きを生じる別の原因がある。VLIW 方式では、演算フィールド数に等しい数の演算命令を 1 つの長語命令に合成できると

3

き最高の処理性能を達成できる。どれだけの演算命令を1つの長語命令に合成することができるかは、コンパイラの性能だけではなく、ソースプログラムの性質にも左右される。ソースプログラムに含まれる並列実行できる演算命令の割合が低ければ、1つの長語命令に合成することができる演算命令数は少ない。一般的なプログラムでは、並列実行できる演算命令はそれほど多くない。例えば、日経エレクトロニクス第487号によれば、平均して2〜3演算命令しか並列に実行できない。

【0004】

【発明が解決しようとする課題】上記のように、VLIW方式には、

(a) 長語命令内に実行時間が異なる演算命令が含まれていると、実行完了を待ち合わせるため、実行部に空きが生じる。

(b) 一般的なプログラムでは、並列性はそれほど高くなく、長語命令内の演算フィールドすべてに演算命令を埋めることができるのはまれである。

といった短所がある。本発明の目的は、

(1) 実行完了待ち合わせにともなう実行部の空きを解消する。

(2) 実効並列度を上げる。

手段を提供することにある。

【0005】

【課題を解決するための手段】上記目的を達成するため、本発明は、演算の種類毎に分類されたnフィールド(nは1以上)の演算フィールドを有し、演算フィールド内の各演算命令が他の演算フィールド内の演算命令と互いに依存関係が無いように構成された長語命令からなる命令流m本(mは1以上)の処理を行なう情報処理装置であって、m個の命令デコード部と、該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられたn個の命令スケジュール部と、該n個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール部から発行された演算命令を実行する実行部と、レジスタ部を備え、前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間におけるレジスタのデータ依存関係およびリソース競合に基づき、実行中の長語命令内の演算命令全ての実行が完了していなくとも前記命令バッファ内の長語命令の前記命令スケジュール部への発行を制御する手段を備えるようにしている。また、m個の命令デコード部と、該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられたn個の命令スケジュール部と、該n個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール部から発行された演算命令を実行する実行部と、レジスタ部を備え、前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間における、レジスタのデ

4

ータ依存関係を判定する判定手段とリソース競合を判定する判定手段と、該両判定手段の判定結果がレジスタのデータ依存関係がなく且つリソース競合がないことを示すとき、実行中の長語命令内の演算命令全ての実行が完了していなくとも命令バッファ内の長語命令を各演算命令に分け対応する命令スケジュール部へ発行制御する手段を備え、前記各命令スケジュール部は、前記命令デコード部から発行された演算命令を保持する命令スケジュールバッファと、命令スケジュールバッファに保持された演算命令の中から実行部に投入する演算命令を選択し、該選択した演算命令を命令流番号と共に実行部に向け投入制御する手段を備え、前記各実行部は、前記命令スケジュール部から投入される演算命令を実行する演算命令実行手段と、命令スケジュール部から投入される命令流番号を受け取りこれを保持する手段と、前記投入された演算命令と命令流番号から、前記演算命令実行手段の動作を制御する手段と、前記演算命令実行手段の状態を前記命令デコード部および命令スケジュール部に通知する手段を備えるようにしている。また、前記実行部の演算命令実行手段は、演算命令の実行により例外発生を検知したとき例外発生信号を前記命令流番号に対応する命令デコード部に通知し、前記デコード部は、例外発生信号を通知した実行部に対して、前記全ての実行部の演算命令実行手段の状態が受け入れ可能状態のとき、例外発生をした演算命令の再実行を指示する手段を備えるようにしている。

【0006】

【作用】上記手段により、実行完了待ち合わせにともなう実行部の空きを解消することができ、実行部における命令実行を効率よく行なうことができる。また、複数命令流の命令をスケジューリングすることにより、実効並列度を上げ、データ依存関係に因る実行部の空きを解消できる。その結果、演算器の空きが生じることなく、高い並列性を得ることができる。

【0007】

【実施例】本発明の実施例を図を用いて説明する。以下、二重引用符“ ”で囲まれた数字は2進数を、引用符で囲まれていない数字は10進数を表すとする。図8は本発明の長語命令の一実施例を示す。本実施例の長語命令は1命令内にL/S161、FAdd162、FMult163、FixOp164の4演算フィールドを持つ。各演算フィールドには、それぞれ以下の演算命令が入る。

L/S演算フィールド : 主記憶とレジスタ間でデータのロード/ストアを行なう演算命令(以下、L/S命令)

FAdd演算フィールド : 浮動小数点加減算を行なう演算命令(以下、FAdd命令)

FMult演算フィールド : 浮動小数点乗算を行なう演算命令(以下、FMult命令)

FixOp演算フィールド：整数演算を行なう演算命令（以下、FixOp命令）

本実施例の演算命令は、主記憶とレジスタの間のデータのやり取りをL/S命令のみに限る、ロード/ストア方式である。

【0008】各演算フィールドの演算命令の持つオペランドの数は、次のとおりである。

FAdd演算フィールド、FMult演算フィールド、FixOp演算フィールド

演算ソース：2

演算ディスティネーション：1

L/S演算フィールド

アドレス計算ソース：1

ロードデータディスティネーション：1（データロード命令の場合）

ストアデータソース：1（データストア命令の場合）。

【0009】本発明の演算命令の実施例を図9に示す。図9（a）はFAdd命令、FMult命令、FixOp命令の構成を、図9（b）は、L/S命令のうちデータロード命令の構成を、図9（c）は、L/S命令のうちデータストア命令の構成を示す。

（a）図9に示すようにFAdd命令、FMult命令、FixOp命令は、オペコード、ディスティネーションレジスタ番号、空きフィールド、ソース1レジスタ番号、ソース2レジスタ番号の各フィールドから構成される。命令の実行は、ソース1レジスタ番号とソース2レジスタ番号のフィールドで示されるレジスタの値に、オペコードフィールドで表される演算を施し、ディスティネーションレジスタ番号フィールドで示されるレジスタに格納する。

（b）データロード命令は、オペコード、ディスティネーションレジスタ番号、オフセット、アドレス計算レジスタ番号、空きフィールドの各フィールドから構成される。命令の実行は、アドレス計算レジスタ番号フィールドで示されるレジスタの値にオフセットフィールドの値を加算し、主記憶のアドレスを求め、このアドレスからデータを、ディスティネーションレジスタ番号フィールドで示されるレジスタに転送する。

（c）データストア命令は、オペコード、空きフィールド、オフセット、アドレス計算レジスタ番号、ストアデータレジスタ番号の各フィールドから構成される。命令の実行は、アドレス計算レジスタ番号フィールドで示されるレジスタの値にオフセットフィールドの値を加算し、主記憶のアドレスを求め、このアドレスにデータを、ストアデータレジスタ番号フィールドで示されるレジスタから転送する。

本実施例の長語命令は、コンパイラがデータの依存関係や制御の依存関係の無い演算命令を組み合わせで合成する。コンパイラが長語命令を合成するとき、適切な演算命令が無く、演算命令を埋め込むことができない場合

は、その演算フィールドにはNop命令が挿入される。Nop命令は何も動作を起こさない演算命令である。

【0010】一実施例として、3本の独立した命令流を処理する情報処理装置を図1に示す。3本の命令流間には、データや制御の依存関係は無い。この3本の命令流を以下、命令流A、命令流B、命令流Cと呼ぶ。本発明の情報処理装置は、主記憶21、命令フェッチ部22、命令デコード部23、命令スケジュール部24、実行部25、レジスタ部26、その他の装置からなる。このうち、図1では、その他の装置は省略してある。主記憶21は、命令流A、命令流B、命令流Cそれぞれのプログラムやデータを格納している。主記憶21には、キャッシュメモリを含むものとする。命令フェッチ部22の働きは、

（1）各命令流毎に、長語命令をフェッチする主記憶のアドレス131を主記憶21に送り、そのアドレスから長語命令132を取り出し、命令デコード部23に送ることである。

【0011】命令デコード部23から命令発行信号133が送られてくれば、毎マシンサイクルごと以上の動作を繰り返す。一実施例として図1に示すような3本の命令流を処理する場合、命令流A、命令流B、命令流Cそれぞれの命令をフェッチするアドレス131は、互いに独立して主記憶21に送られ長語命令が取り出される。命令デコード部23の働きは、

- （1）演算命令発行の可能性を判定する
- （2）命令スケジュール部24に向け演算命令を発行する
- （3）レジスタの状態の管理する
- （4）命令フェッチ部22に次命令のフェッチを要求する

ことである。命令デコード部23は、毎マシンサイクルごと以上の動作を繰り返す。一実施例として図1に示すような3本の命令流を処理する場合、命令デコード部23は、命令流A、命令流B、命令流Cに対応して、命令デコード部A 23a、命令デコード部B 23b、命令デコード部C 23cの3本が設けられている。そして、それぞれの命令デコード部23は干渉することなく独立して動作する。

【0012】命令スケジュール部24は各命令毎に設けられ、命令スケジュール部24の働きは、

（1）命令デコード部23から発行された演算命令135のなかから、実行部25に投入する演算命令を選択することである。命令スケジュール部24は、命令デコード部23より命令が発行されているならば、毎マシンサイクルごとに以上の動作を繰り返す。一実施例として図8に示すような4つの演算フィールドを有する長語命令の場合、命令スケジュール部24は、各演算フィールドに対応して、L/S命令用の命令スケジュール部24

a、FAdd命令用の命令スケジュール部24b、FM

u l t 命令用の命令スケジュール部 2 4 c、F i x O p 命令用の命令スケジュール部 2 4 d の 4 本が設けられている。そして、それぞれの命令スケジュール部 2 4 は干渉することなく独立して動作する。

【0013】実行部 2 5 は各命令毎に設けられ、実行部 2 5 の働きは、

(1) 命令スケジュール部 2 4 から投入される演算命令を実行する

(2) 次の演算命令を受け入れ可能であるかを判定することである。

【0014】実行部 2 5 は、演算パイプライン化されており、先行演算命令の実行が完了せずとも、次の演算命令実行を開始できる。しかし、演算命令の種類により、また、実行状態により、次の演算命令がいつ実行開始できるかは異なる。そのため、実行部 2 5 が、演算命令の実行状態をみて、次の演算命令を受け入れ可能であるか、あるいは、不可能であるかを判定する。実行部 2 5 は、命令スケジュール部 2 4 から演算命令が投入されている間は以上の動作を毎マシサイクルごと繰り返す。一実施例として図 8 に示すような 4 つの演算フィールドを有する長語命令の場合、実行部 2 5 は、各演算フィールドに対応して、L/S 命令用の実行部 2 5 a、F A d 命令用の実行部 2 5 b、F M u l t 命令用の実行部 2 5 c、F i x O p 命令用の実行部 2 5 d の 4 本が設けられている。そして、それぞれの命令実行部 2 5 は干渉することなく独立して動作する。

【0015】レジスタ部 2 6 の働きは、

(1) データを保持する

(2) レジスタの状態を保持する

(3) レジスタの状態を更新する

ことである。レジスタの状態は、

(a) 未定

(b) 確定

の 2 値をとる。

(a) 未定状態とは、レジスタの値を書き換える演算命令が命令デコード部 2 3 から発行されたが、未だ実行が完了しておらず、したがって、レジスタの値が未定な状態である。

(b) 確定状態とは、演算命令が完了し、レジスタの値がすでに確定している状態である。

一実施例として図 1 に示すような 3 本の命令流を処理する場合、レジスタ部 2 6 は、各命令流に対応して、2 6 a、2 6 b、2 6 c の 3 本が設けられている。そして、それぞれのレジスタ部 2 6 は干渉することなく独立して動作する。

【0016】命令フェッチ部 2 2 の動作をさらに詳しく説明する。ある命令に注目してみた場合、命令フェッチ部 2 2 は、主記憶 2 1 に命令フェッチアドレス 1 3 1 を与え、主記憶 2 1 から長語命令を取り出し、信号線 1 3 4 を使って命令デコード部 2 3 内の命令バッファ 3 1

(図 2) に送る。先行長語命令が命令デコード部 2 3 から発行されると次の長語命令を主記憶 2 1 からフェッチする。命令デコード部 2 3 から長語命令が発行されたかどうかは、命令発行信号 1 3 3 に示される。

命令発行信号 1 3 3 が“1”

のとき、命令デコード部 2 3 から長語命令が発行され、命令フェッチ部 1 1 は命令フェッチを行なう。

命令発行信号 1 3 3 が“0”

ならば、命令デコード部 2 3 から長語命令が発行されず、したがって、命令フェッチも行なわない。以上は、各命令流ごと独立して行なわれる。

【0017】次に、命令デコード部 2 3 の動作をさらに詳しく説明する。図 2 に命令デコード部 2 3 の一実施例を示す。図 2 中、命令デコード部 2 3 は、命令バッファ 3 1、競合判定部 3 2、依存関係解明部 3 3、実行制御部 3 4、受け入れ可能フラグ 3 5、命令発行器 3 6 から構成される。命令デコード部 2 3 を構成する各部の動作を以下に説明する。命令バッファ 3 1 は、命令フェッチ部 2 2 から送られてくる長語命令を各命令スケジュール部 2 4 に向け発行するまでの間保持する。

【0018】競合判定部 3 2 は実行部 2 5 の競合状態を調べる。図 10 に競合判定部 3 2 の一実施例を示す。実行部 2 5 の競合状態とは、(a) 命令バッファ 3 1 内の長語命令に N o p 命令でない演算フィールドが有り、かつ、(b) その、演算フィールドに対応する実行部 2 5 に演算命令の受け入れが不可能なものが 1 以上存在する状態である。各実行部 2 5 の演算命令受け入れ可能であるか不可能であるかは、受け入れ可能信号 1 4 3 が書き込まれた受け入れ可能フラグ 3 5 に示される。競合判定部 3 2 は、受け入れ可能フラグ 3 5 (受け入れ可能なら“1”、そうでなければ“0”)と、命令バッファ 3 1 の各演算命令と N o p 命令とを比較した比較結果 (演算命令が N o p 命令なら“1”、そうでなければ“0”)から、競合状態を調べ、信号線 1 5 1 に結果を出力する。競合判定出力 1 5 1 の値は、

実行部 4 本全てが競合状態でない：競合判定出力 1 5 1 = “1”

実行部 4 本のうち 1 本でも競合状態：競合判定出力 1 5 1 = “0”

である。

【0019】依存関係解明部 3 3 は、実行中の演算命令と命令バッファ 3 1 内の長語命令間で、レジスタのデータ依存関係を調べる。図 11 に依存関係解明部 3 3 の一実施例を示す。データ依存関係には、

(a) RAW の依存関係

(b) WAR の依存関係

(c) WAW の依存関係

がある。まず、以上 3 種類のデータ依存関係について説明する R 1 ~ R 5 はレジスタを表すとする。R 1 < - R 2 + R 3 は、R 2 と R 3 の値を加算して、その結果を

R1に代入することを表す。

(a) RAW (Read After Write) の依存関係

1: R1 ← R2 + R3

2: R4 ← R1 + R5

1、2の順で命令を実行した場合、R1の値が確定するまで、2の命令は実行開始することができない。

(b) WAR (Write After Read) の依存関係

1: R2 ← R1 + R3

2: R1 ← R4 + R5

1、2の順で命令を実行した場合、1の命令が実行開始した後でないと、2の命令の結果をR1に書き込めない。

(c) WAW (Write After Write) の依存関係

1: R1 ← R2 + R3

2: R1 ← R4 + R5

1、2の順で命令を実行した場合、1の命令が結果を書き込んだ後でないと、2の命令の結果をR1に書き込めない。以上、3通りの依存関係のうち、WARの依存関係は生じない。それは、先行する長語命令内の演算命令全てが実行開始した後、すなわち、ソースレジスタの値を読み出した後でないと、次の長語命令を命令デコード部23から発行しないためである。よって、依存関係解明部33が解明しなければならないデータ依存関係は、

(a) RAWの依存関係、(c) WAWの依存関係である。この、2種類の依存関係は、レジスタ値が書き込まれる前に、(a) レジスタの読みだし、(c) レジスタの書き込みを行なうと生じる。すなわち、未定状態のレジスタを使用しようとすると、(a) RAWの依存関係、(c) WAWの依存関係は生じる。よって、依存関係解明部33では、レジスタの状態が未定状態であるかを調べれば、データ依存関係を解明できる。レジスタの状態は、レジスタ部26が持つ予約ビット174 (図6) に示される。未定状態のレジスタの予約ビット174の値は“1”である。命令バッファ31に保持された演算命令のオペランドの各レジスタの予約ビット174を調べ、依存関係解明出力152に出力する。依存関係解明出力152の値は、

全ての予約ビットが“0”：データ依存関係無、
依存関係解明出力152 = “1”

“1”の予約ビットが存在：データ依存関係有、
依存関係解明出力152 = “0”

である。

【0020】実行制御部34は、各実行部25の実行状態を監視し、自命令流の演算命令により例外が発生したならば、例外処理を行なう。他の命令流の演算命令が例外を発生された場合、何もしない。図12に実行制御部34の一実施例を示す。実行制御部34は、例外発生フ

ラグ53と再実行指示信号生成論理54から構成される。各実行部25で例外(0で除算した場合、オーバーフローが発生した場合等)が発生すると、実行部25は、対応する命令流の命令デコード部23内の実行制御部34に、例外が発生したことを例外発生信号141を通じ報告する。例外発生信号141を受け取った実行制御部34は、例外発生フラグ53に書き込む。例外発生フラグ53は一実施例として図1に示すような4つの実行部を有する場合、4bitで構成され、それぞれのビットが各実行部25に対応する。例外処理は、演算例外を起こした演算命令を含む長語命令内の全ての演算命令の実行が全て終了した後、開始する。全ての演算命令の実行が終了したか／否かは、受け入れ可能フラグ35の全ビットのAND(論理積)をとった信号線143-1の値を基に検出する。再実行指示信号生成論理54は、実行部25に対し演算命令の再実行を指示する信号142を生成する。例外発生フラグ53の値と、信号線143-1の値のAND(論理積)をとることで、再実行指示信号142を生成する。再実行開始時、対応する演算フィールドの再実行指示信号142は、再実行指示信号142 = “1”

になる。例外処理開始待機中あるいは例外処理実行中であるか、正常状態であるかは、実行状態出力153に出力される。実行状態出力153の値は、

正常状態：実行状態出力153 = “1”

例外処理開始待機中／例外処理実行中：実行状態出力153 = “0”

である。実行状態出力153は、例外発生フラグ53全ビットのNOR(否定論理和)を取ることで生成される。例外発生報告を受けた後、例外処理開始を待っている間、また、例外処理中は、信号線153 = “0”

になり、例外処理開始待機中／例外処理実行中を示す。その結果、発行条件が整っていても命令バッファ31内の長語命令は発行されない。

【0021】受け入れ可能フラグ35は、実行部25が演算命令を受け入れ可能であるかどうかを示すフラグで、一実施例として図1に示すような4つの実行部を有する場合、4bitで構成され、各ビットがそれぞれの実行部に対応している。受け入れ可能フラグ35の各ビットの値は、

受け入れ可能： “1”

受け入れ不可能： “0”

である。受け入れ可能フラグ35は、各実行部25が、演算命令の実行状態に応じ書き換える。

【0022】命令発行信号133は、命令バッファ31から長語命令が発行されることを示す信号で、長語命令が発行される場合“1”

になる。長語命令が発行される条件、すなわち、命令発行信号133が“1”になる条件は、

(a) 競合関係がなく(競合判定出力151="1")
かつ、
(b) データ依存関係がなく(依存関係解明出力152="1")
かつ、
(c) 正常状態(実行状態出力153="1")
である。命令発行信号133が"1"のとき、命令発行器36が開き、命令バッファ31内の長語命令が演算命令に分けられて命令スケジュール部24に向け発行される。また、命令発行信号133の値は命令フェッチ部22にも送られ、次の長語命令のフェッチを指示する。命令発行器36は、命令発行信号133の値に基づき、命令バッファ31内の長語命令を命令スケジュール部24に向け発行する。命令発行器36は、命令発行信号114が"1"のとき、命令バッファ31内の長語命令を命令スケジュール部24に向け発行する。信号線140は、発行された演算命令の結果格納先レジスタの予約ビットを、
予約ビット="1"
にすることを、レジスタ部26に指示する信号である。信号線140は、演算命令発行時に、
信号線140="1"
になる。信号線140は、命令発行信号133と同じ条件で生成される。

【0023】次に、命令スケジュール部24の働きについて詳しく説明する。図3に命令スケジュール部の一実施例を示す。命令スケジュール部24は、スケジュールバッファ37、Opecodeコンパレータ40、スレッド選択論理38、2bitカウンタ(一実施例として図1に示すように3命令流の場合)39、投入演算命令セクタ41からなる。スケジュールバッファ37は、各命令デコード部23から発行された演算命令を、実行*

TNC	: 命令流番号カウンタ39の値
STN	: 開始時の命令流番号カウンタ39の値を保持する作業用のレジスタ
Signal141	: 例外発生信号141の値
Signal143	: 受入れ可能信号143の値
Signal154	: スレッド選択信号154の値
Signal155(n)	: 命令流番号nの命令流の信号線155の値
A<-B	: AにBの内容を転送する動作を示す
A==B	: AとBの値が等しいかどうかを判定

図13のスレッド選択信号154生成手順を説明する。左端カッコ付きの数字は、図13のカッコ付きの数字に対応する。

【0027】(1): Signal143=="0"
(実行部が新たな演算命令を受入れ不可能)

あるいは

Signal141=="0" (例外発生中)

ならば、スレッド選択信号の生成を開始せず、開始できるまで待ち合わせる。

*部25に投入するまで保持する。スケジュールバッファ37は、一実施例として図1に示すように3命令流の場合、3エン트리から構成される。例えば、命令スケジュール部がL/S命令用の命令スケジュール部24aの場合、各エントリには各命令流のL/S命令あるいはNop命令が格納される。Opecodeコンパレータ40は、スケジュールバッファ内の演算命令がNop命令であるかどうかを判定し、信号線155に出力する。信号線155の値は、

10 Nop命令 : 信号線155="1"

Nop命令以外: 信号線155="0"

である。一実施例として図1に示すような3本の命令流を処理する場合、Opecodeコンパレータ40は各命令流に対応して3本設けられている。

【0024】スレッド選択論理38は、

(a) 投入する演算命令の選択

を行なう。図13にスレッド選択論理の一実施例を示す。

(a) 投入する演算命令の選択は、2bitカウンタ39の値を基に行なわれ、投入演算命令セクタ41を制御するスレッド選択信号154を、2bitカウンタ39の値から生成する。一実施例として図1に示すような3本の命令流を処理する場合、各命令流を識別する命令流番号をそれぞれ、

"00" (命令流A)、

"01" (命令流B)、

"10" (命令流C)

とする。"11"は、選択する命令流がないことを表す。

【0025】スレッド選択信号154の生成手順を図13に示す。図13中の記号の意味は次ぎのとおりである。

【0026】

(1): STNにTNCを転送する。STNはスレッド

選択信号154生成を終了させるのに使用する。

(2): Signal155(TNC)="0"ならば、

TNCが示す命令流番号のオペコードはNop命令ではない。従って、TNCの示す命令流を選択する。

(3): スレッド選択信号154の値(Signal154)をTNCの値にし、TNCの値を更新する。

(4): TNCを更新した結果、TNC="11"ならば、

TNCを"00"にして終了。TNC="11"でな

50 ば、TNCを"00"にして終了。TNC="11"でな

51 ば、TNCを"00"にして終了。TNC="11"でな

52 ば、TNCを"00"にして終了。TNC="11"でな

53 ば、TNCを"00"にして終了。TNC="11"でな

54 ば、TNCを"00"にして終了。TNC="11"でな

れば、そのまま終了。

(6) : (3) で、Signal 155 (TNC) = "0" でなければ、TNC が示す命令流番号のオペコードは Nop 命令。TNC を更新して、再びオペコードを調べる。

(7) : TNC を更新した結果、TNC = "11" ならば、TNC を "00" にする。

(8) : STN = TNC ならば、TNC の値が一回りした。

【0028】このときは、TNC を "11" (選択命令流無し) スレッド選択信号 154 の値を "11" (選択命令流無し)

にして終了。投入演算命令セクタ 41 は、スレッド選択論理 38 からの命令流選択信号 154 により実行部 25 に投入する演算命令を選択する。命令流選択信号 154 が示す命令流番号の演算命令を実行部 25 に投入するようにセクタが開く。

【0029】図 4～図 5 に実行部 25 の一実施例を示す。図 4 に示すのは、メモリアクセスを伴わない演算を実行する実行部 25 の実施例である。また、図 5 に示すのは、メモリアクセスをとともう演算命令を実行する実行部 25 の実施例である。メモリアクセスを伴わない演算を実行する実行部 25 から説明する。図 4 に示す実行部 25 は、演算器 42、命令流番号レジスタ 44、演算制御器 43 からなる。演算器 42 は、実行部 25 に投入された演算命令 137 にしたがってレジスタ部 26 とデータをやり取りし、演算を行なう。一実施例として図 4 に示す実行部 25 の場合、演算器 42 は 2 stage の演算パイプライン化されている。演算器 42 は、演算の段階ごとに実行状態を信号線 156 を通じ演算制御器 43 に送る。また、演算の各段階の制御も信号線 156 を通じ演算制御器 43 から送られる。命令流番号レジスタ 44 は、実行中の演算命令の命令流番号を保持している。命令流番号レジスタ 44 に保持された命令流番号は、

- (a) 演算実行に伴うレジスタ部 26 のアクセス
- (b) 命令デコード部 23 に対する例外発生 の通知

に使用される。演算制御器 43 は、

- (a) 演算器 42 の制御
- (b) 例外発生 の検出と通知
- (c) 例外処理の制御

を行なう。

(a) 演算制御器 43 は、命令スケジュール部 24 から投入された演算命令 137 を実行するために、信号線 156 を使って演算器 42 を制御する。演算命令実行の各段階の演算器 42 の状態を信号線 156 を通じ、演算制御器 43 は知る。演算命令 137 の実行が進み、新たな演算命令を受け入れることが可能になったならば、受け入れ可能信号 143 = "1"

にして、命令スケジュール部 24、命令デコード部 23

に知らせる。命令デコード部 23 に送られた信号線 143 は、命令デコード部 23 内の受け入れ可能フラグ 35 の値を書き換える。

(b) 演算命令を実行中、演算制御器 43 が例外を検出すると、

例外発生信号 141 = "1"

にして、命令デコード部 23 に例外の発生を知らせる。

演算制御部 43 が例外を通知する命令デコード部 23 は、命令流レジスタ 44 に保持された命令流番号により特定される。

(c) 命令デコード部 25 から再実行指示信号 142 を受け取ると、演算制御部 43 は例外を発生させた演算命令の再実行をする。再実行が完了したならば、

受け入れ可能信号 143 = "1"

にして、命令スケジュール部 24、命令デコード部 23 に再実行の完了を知らせる。

【0030】次に、図 5 に示す実行部について説明する。図 4 に示す実行部の実施例と違い、メモリアクセスのために、メモリアクセス制御器 47 が設けられている。また、アドレス演算器 45 は、図 4 に示す実行部 42 の実施例とは異なり、アドレス演算を行なう。アドレス演算器 45 はメモリアクセスアドレス 146 を計算するために使用される。アドレス演算器 45 は、演算命令 137 のアドレス計算オフセット 169 (図 9) の値と、アドレス計算ソース 144 から、メモリアクセスアドレス 146 を演算する。アドレス演算器の出力 (メモリアクセスアドレス 146) は、メモリアクセス制御器 47 に入力する。実行制御器 46 は、メモリアドレスを計算するよう信号線 156 を通じてアドレス演算器 45 に指示を出し、同時に、信号線 157 を通じ、メモリアクセス制御器 47 に、メモリアクセスの種類を知らせる。メモリアクセスの状態は、信号線 157 を通じ実行制御器 46 に知らされる。実行制御器 46 に伝えられるメモリアクセスの状態には、例として、キャッシュメモリのヒット/ミスヒット、がある。

(a) 演算命令 137 がデータロード命令の場合のメモリアクセス制御器 47 の動作を説明する。

1) メモリアクセスアドレス 146 を信号線 121 を通じ主記憶 21 に出力

2) データが転送されてくるのを待つ

3) 障害が発生したならば、信号線 157 を通じ実行制御部 46 に知らせる。障害の一例としてキャッシュメモリのミスヒットがある。

4) データが信号線 120 を通じ転送されてきたならば、信号線 145 を通じ、転送されてきたレジスタ部 26 に書き込む

5) レジスタ部 26 にデータを書き込んだ段階で、

受け入れ可能信号 143 = "1"

にする。

(b) 演算命令 137 がデータストア命令の場合のメモ

リアクセス制御器47の動作を説明する。

- 1) メモリアクセスアドレス146を信号線121を通じ主記憶21に出力
- 2) 続いて、ストアデータ144を信号線121を通じ主記憶21に出力
- 3) 障害が発生したならば、信号線157を通じ実行制御器46に知らせる。障害の一例としてキャッシュメモリのミスヒットがある。
- 4) 主記憶21にデータを書き込んだ段階で、受け入れ可能信号143="1"にする。

【0031】レジスタ部26についてさらに詳しく説明する。図6にレジスタ部26の一実施例を示す。レジスタ部内の各レジスタは、

- (1) データ領域173
- (2) 予約ビット174
- (3) ソース1レジスタ番号保持ラッチ175
- (4) ソース2レジスタ番号保持ラッチ176
- (5) ディスティネーションレジスタ番号保持ラッチ177

から構成される。データ領域173は、レジスタが持つ値を格納しておく領域である。予約ビット174は、レジスタの状態を表すビットで、図6に示す実施例では、1bitで構成されている。予約ビット174の値は、未定：予約ビット="1"
確定：予約ビット="0"

である。予約ビット174の値は、データの依存関係解明のため命令デコード部23内の依存関係解明部33から読み出される。

【0032】ソース1レジスタ番号保持ラッチ175は、レジスタ番号139に示されるレジスタ番号のうち、ソース1レジスタ番号フィールド167、あるいは、アドレス計算レジスタ番号フィールド170のレジスタ番号を保持する。ソース1レジスタ番号フィールド167、あるいは、アドレス計算レジスタ番号フィールド170が演算命令にない場合は、'Invalid number'が保持される。'Invalid number'はレジスタ番号としては無意味な数である。ソース2レジスタ番号保持ラッチ176は、レジスタ番号139に示されるレジスタ番号のうち、ソース2レジスタ番号フィールド168、あるいは、ストアレジスタ番号フィールド171のレジスタ番号を保持する。ソース1レジスタ番号フィールド167、あるいは、アドレス計算レジスタ番号フィールド170が演算命令にない場合は、'Invalid number'が保持される。ディスティネーションレジスタ番号保持ラッチ177は、レジスタ番号139に示されるレジスタ番号のうち、ディスティネーションレジスタ番号フィールド166のレジスタ番号を保持する。ディスティネーションレジスタ番号フィールド166が演算命令にない場合

は、'Invalid number'が保持される。レジスタ部26を構成する各1本のレジスタは、ソース1レジスタ番号保持ラッチ175、ソース2レジスタ番号保持ラッチ176、ディスティネーションレジスタ番号保持ラッチ177に保持されたレジスタ番号により特定される。

【0033】演算命令が発行されると信号線140が有効になる。信号線140が有効になったならば、レジスタ部26はディスティネーションレジスタ番号保持ラッチ177に保持されたレジスタの予約ビット174の値を、予約ビット174="1"

にして、レジスタの状態を未定状態にする。演算命令の実行完了に伴い、演算結果が信号線145を通じ書き込まれたならば、レジスタ部26はディスティネーションレジスタ番号保持ラッチ177に保持されたレジスタの予約ビットの値を、予約ビット="0"

にして、レジスタの状態を確定状態にする。ソース1レジスタ番号保持ラッチ175、ソース2レジスタ番号保持ラッチ176、ディスティネーションレジスタ番号保持ラッチ177は、一実施例として図8に示すような4演算フィールドからなる長語命令を処理する場合、それぞれ4組づつ設けられている。

【0034】次に長語命令がどのように処理されるかを説明する。一実施例として図1に示すような、命令流A、命令流B、命令流Cの3本の命令流を処理する場合を例に取り上げる。以下の説明で、左端の数字は、命令デコード部23aが命令発行信号133aを命令フェッチ部22に送った時点から計ったマシンサイクル数を表す。

00 命令デコード部23aが命令発行信号133aを命令フェッチ部22に送る。

01 命令発行信号133aを受け取った命令フェッチ部は22は、命令フェッチアドレス131aを主記憶21に送り、長語命令がフェッチされてくるのを待つ。

02 キャッシュにヒットしたならば、この時刻に長語命令が命令フェッチ部22に取り込まれる。キャッシュがミスヒットしたならば、長語命令が取り込まれるまで待合せを行なう。以下の処理は、キャッシュのヒット/ミスヒットに関わりなく同じであるため、キャッシュにヒットしたとして説明を続ける。

03 命令フェッチ部22に取り込まれた長語命令は、命令デコード部23a内の命令バッファ31に転送される。

【0035】04 命令バッファ31内の長語命令は、

競合判定部32、

依存関係解明部33

に送られ、実行部25の競合判定と、レジスタの依存関

係が調べられる。依存関係解明部 33 に送られると同時に、長語命令から、レジスタ番号が取り出され、信号線 139 を通じ、レジスタ部 26a に送られる。レジスタ部 26a は、ソース 1 レジスタ番号保持ラッチ 175、ソース 2 レジスタ番号保持ラッチ 176、ディスティネーションレジスタ番号保持ラッチ 177 にそれぞれの演算フィールドごと分けて記憶する。さらに、レジスタ部 26a 信号線 139 を通じ送られてきた前レジスタの予約ビット 174 の値を、信号線 138 を通じ、命令デコード部 23a に送る。この時刻に、

(a) 競合関係がなく (競合判定出力 151 = "1")

かつ、

(b) データ依存関係がなく (依存関係解明出力 152 = "1")

かつ、

(c) 正常状態 (実行状態出力 153 = "1")

であれば、

命令発行信号 133 = "1"

*

命令流カウンタ 39 の値 (TNC)	: "10"
例外発生信号 141 の値 (Signal 141)	: "0"
受け入れ可能信号 143 の値 (Signal 143)	: "1"
Signal 155 ("10")	: "1"
Signal 155 ("00")	: "0"

以下、左端のカッコ付きの数字は図 13 中のカッコ付きの数字に対応する。

(1) Signal (141) = "0" かつ、Signal (143) = "1" であるから、

(2) STN に TNC の値を転送する。

(3) Signal 155 ("10") = "1" より、

(6) TNC を TNC + 1 とする。

(7) TNC = "11" であるから、TNC の値は、"00"

(8) STN ("10") の値と TNC ("00") の値が違うので、

(3) Signal 155 ("00") を評価

Signal 155 ("00") = "00" より、

(4) Signal 154 に TNC ("00") を転送
TNC の値は "01" に更新される。

(5) TNC = "11" でないので、TNC の値はそのまま。

スレッド選択信号 154 の値は、"00" であるので命令流 A が選択される。

【0037】07 投入演算命令セクタ 41 が開き、命令流 A の演算命令が実行部 25b 内の演算制御部 45 に向け送られる。同時に命令流番号 ("00") も実行部 25b 内の命令流番号レジスタ 44 に送られる。

08 命令流番号レジスタ 44 に記憶された命令流番号 ("00") をもとに、レジスタ部 26a を選択し、演算に必要なデータを実行部 25b に転送する。

09 演算を開始する。演算器 42 は演算パイプライン

*になる。

05 命令発行信号 133 = "1"

ならば、命令発行器 36 が開き、命令バッファ 31 内の長語命令が、演算命令に分割され、各命令スケジュール部 24 内のスケジュールバッファ 37 に向け発行される。同時に、命令発行信号 133a は命令フェッチ部 11 に送られ、次のマシンサイクルで、命令流 A の次の長語命令をフェッチする。さらに、予約化信号 140 がレジスタ部 26a に送られ、ディスティネーションレジスタ番号保持ラッチに記憶されたレジスタの予約ビット 174 を未定状態にする。

【0036】06 命令スケジュール部 24a ~ d の処理はどれでも同じである。そこで、命令スケジュール部 24b の動作について述べる。図 13 に示す手順により生成されるスレッド選択信号 154 を基に、スケジュールバッファ 37 内の演算命令は選択される。図 13 中の変数の値を以下のように仮定する。

ン化されているため、新たな命令が受け入れ可能になり、信号線 143 = "1" になる。

10 演算実行中。新たな命令が命令スケジュール部 24b から送られてくる。

11 演算終了。演算結果をレジスタ部 26 に戻す。

レジスタ部 26a はディスティネーションレジスタ番号保持ラッチに記憶されたレジスタのデータ領域に演算結果を書き込むと同時に予約ビットを確定状態にする。

一実施例として図 1 に示すような 3 本の命令流を処理する場合、以上のように長語命令は実行される。

【0038】

【発明の効果】本発明によれば、命令実行の並列性低下の原因となる、データ依存関係および演算器の競合を始めとしたリソース競合を回避でき、その結果実行部が演算命令を実行していない空き時間を減少させ、高い効率で実行部を使用することができる。

40 【図面の簡単な説明】

【図 1】本発明の一実施例を示す図である。

【図 2】命令デコード部の一実施例を示す図である。

【図 3】命令スケジュール部の一実施例を示す図である。

【図 4】実行部の一実施例を示す図である。

【図 5】メモリアクセスを行なう実行部の一実施例を示す図である。

【図 6】レジスタ部の一実施例を示す図である。

【図 7】従来の VLIW 方式の情報処理装置を示す図である。

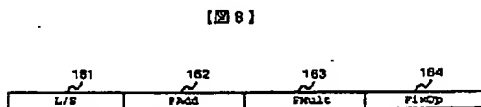
19

【図 8】 本発明の長語命令の一実施例を示す図である。
 【図 9】 本発明の長語命令を構成する演算命令の一実施例を示す図である。
 【図 10】 競合判定部の一実施例を示す図である。
 【図 11】 依存関係解明部の一実施例を示す図である。
 【図 12】 実行制御部の一実施例を示す図である。
 【図 13】 スレッド選択論理がスレッド選択信号を生成する手順を示す図である。

【符号の説明】

21 主記憶
 22 命令フェッチ部
 23 a 命令流 A 命令デコード部
 23 b 命令流 B 命令デコード部
 23 c 命令流 C 命令デコード部
 24 a L/S 命令スケジュール部
 24 b FAdd 命令スケジュール部
 24 c FMult 命令スケジュール部
 24 d FixOp 命令スケジュール部
 25 a L/S 命令実行部
 25 b FAdd 命令実行部
 25 c FMult 命令実行部
 25 d FixOp 命令実行部
 26 a 命令流 A レジスタ部
 26 b 命令流 B レジスタ部
 26 c 命令流 C レジスタ部
 31 命令バッファ
 32 競合判定部
 33 依存関係解明部
 34 実行制御部
 35 受け入れ可能フラグ
 36 命令発行部
 37 スケジュールバッファ
 38 スレッド選択論理

【図 8】

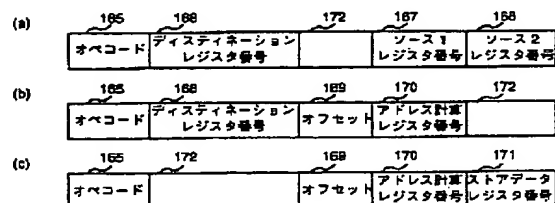


20

39 命令流番号カウンタ
 40 a ~ 40 d opcode コンパレータ
 41 投入演算命令セクタ
 42 演算器
 43 演算制御器
 44 命令流番号レジスタ
 45 アドレス演算器
 46 実行制御器
 47 メモリアクセス制御器
 10 50 レジスタ番号バッファ
 51 a L/S 演算フィールドレジスタ番号バッファ
 51 b FAdd 演算フィールドレジスタ番号バッファ
 51 c FMult 演算フィールドレジスタ番号バッファ
 51 d FixOp 演算フィールドレジスタ番号バッファ
 52 依存関係解明論理
 53 例外発生フラグ
 54 再実行指示信号生成論理
 20 71 従来の VLIW 方式情報処理装置の主記憶
 72 従来の VLIW 方式情報処理装置の命令フェッチ部
 73 従来の VLIW 方式情報処理装置の命令デコード部
 75 a ~ d 従来の VLIW 方式情報処理装置の演算器
 76 従来の VLIW 方式情報処理装置のレジスタ部
 81 従来の VLIW 方式情報処理装置の命令バッファ
 85 従来の VLIW 方式情報処理装置の受け入れ可能ビット
 30 86 従来の VLIW 方式情報処理装置の命令発行部
 175 ソース 1 レジスタ番号保持ラッチ
 176 ソース 2 レジスタ番号保持ラッチ
 177 ディスティネーションレジスタ番号保持ラッチ

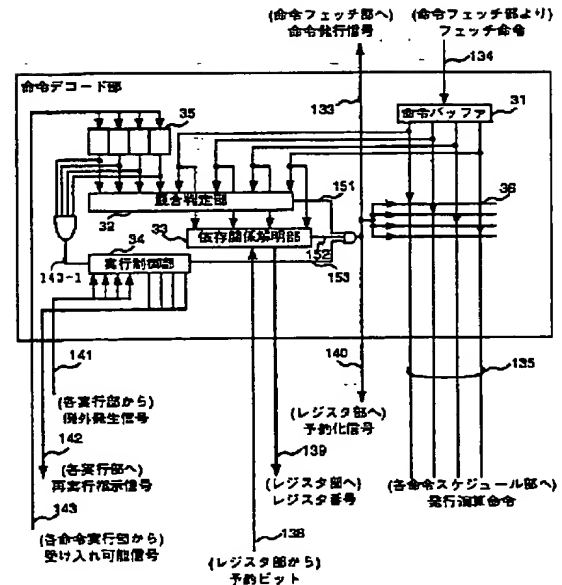
【図 9】

【図 9】



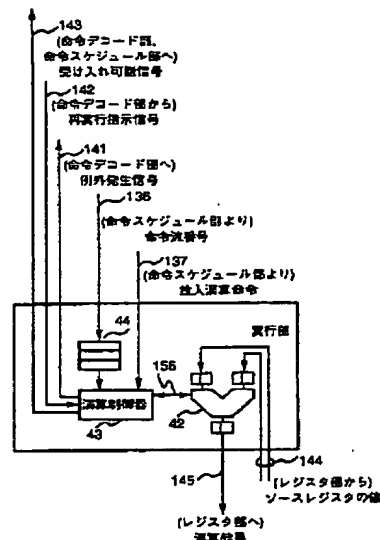
【図2】

[REDACTED]



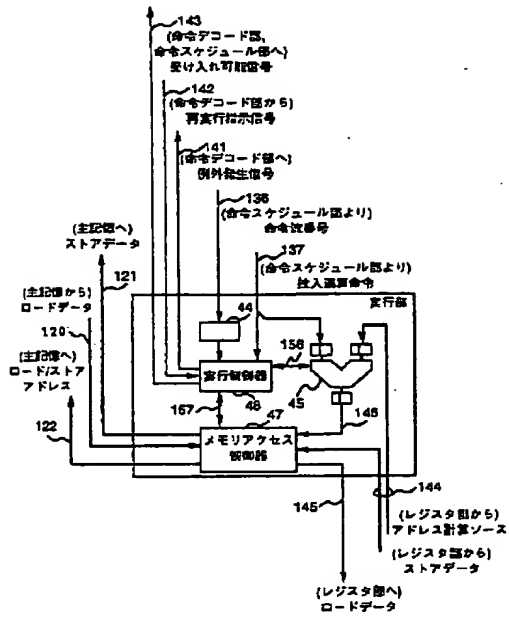
【図4】

[50 4]



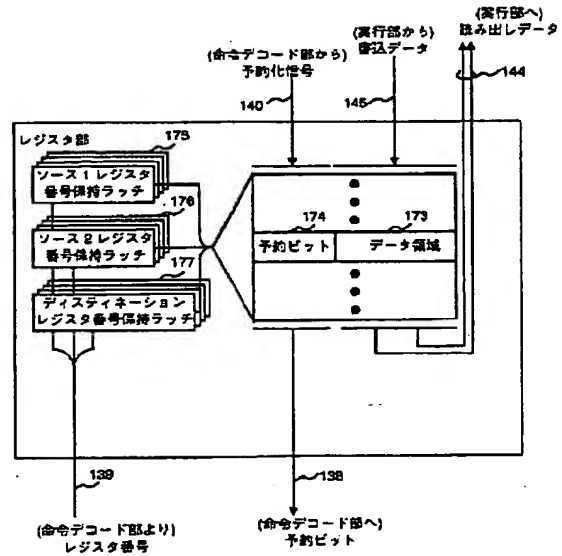
【図 5】

【図 5】



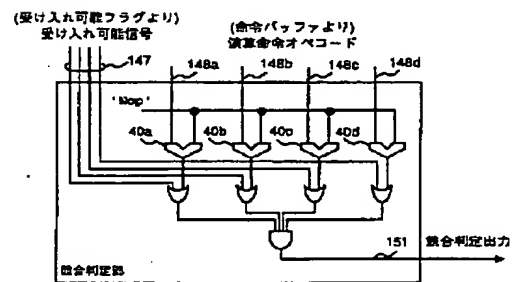
【図 6】

【図 6】



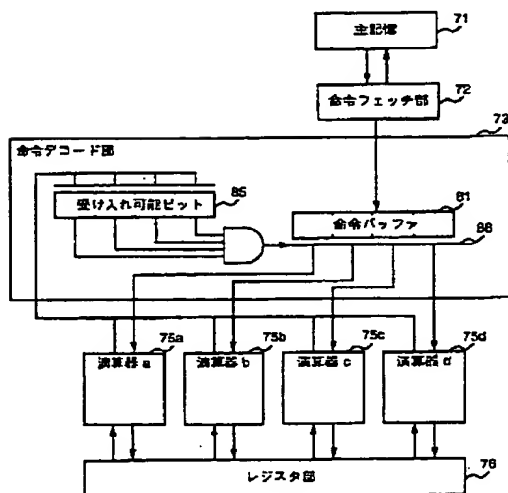
【図 10】

【図 10】



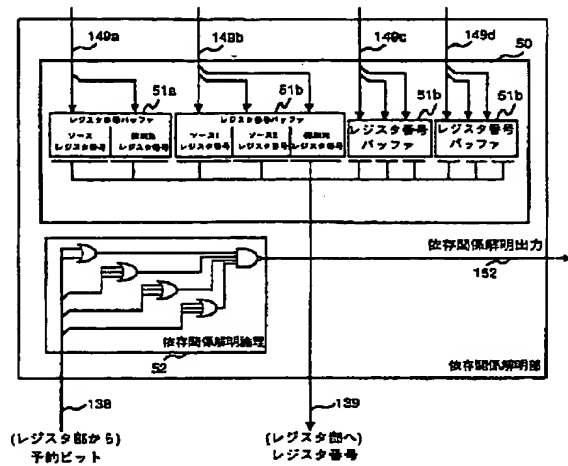
【図 7】

【図 7】



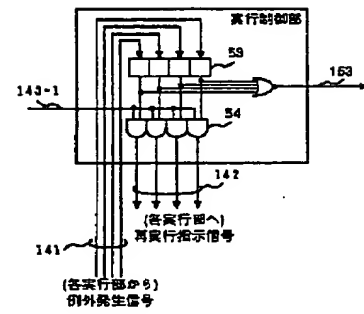
【図 11】

【図 11】



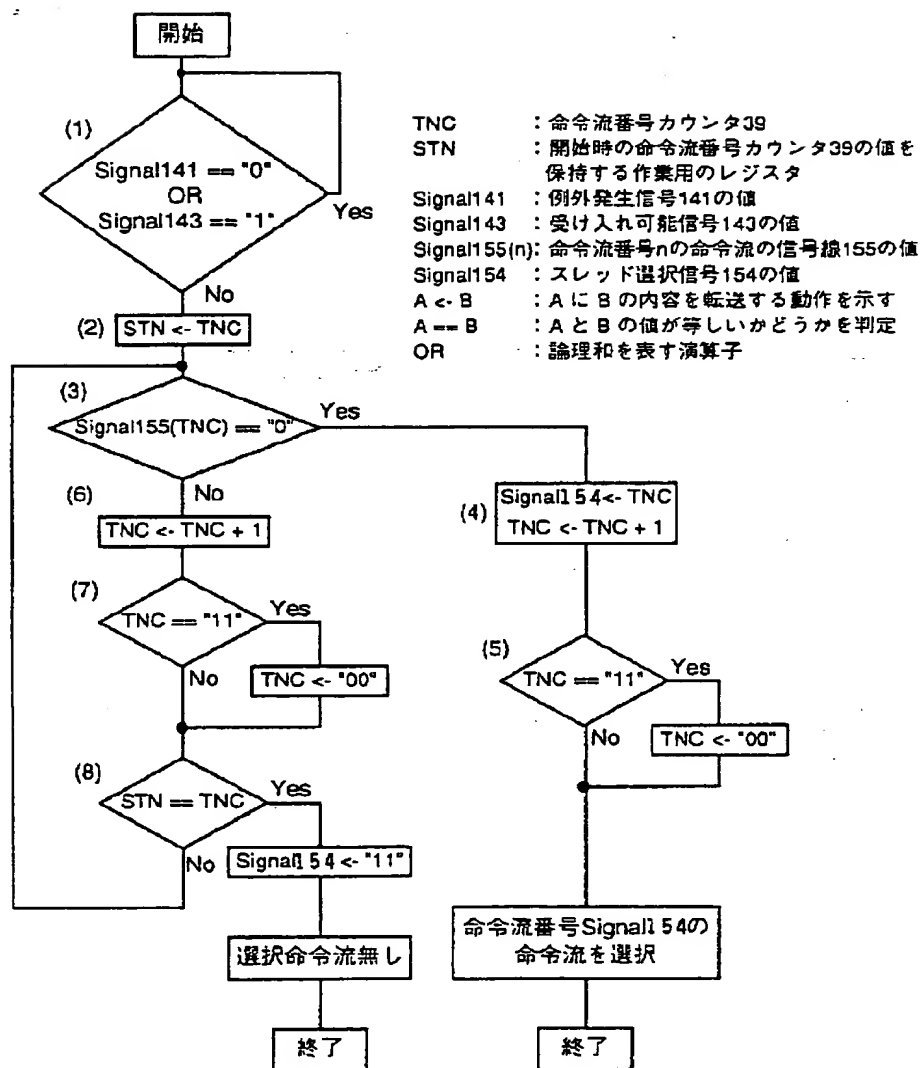
【図 12】

【図 12】



【図13】

【図13】



【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第3区分
【発行日】平成13年11月22日(2001.11.22)

【公開番号】特開平7-281896
【公開日】平成7年10月27日(1995.10.27)
【年通号数】公開特許公報7-2819
【出願番号】特願平6-102275
【国際特許分類第7版】
G06F 9/38 370
【FI】
G06F 9/38 370 B

【手続補正書】

【提出日】平成13年4月11日(2001.4.11)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 演算の種類毎に分類された n フィールド(n は1以上)の演算フィールドを有する長語命令からなる命令流 m 本(m は1以上)の処理を行なう情報処理装置であって、
 m 個の命令デコード部と、
該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられた n 個の命令スケジュール部と、
該 n 個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール部から発行された演算命令を実行する実行部と、
レジスタ部を備え、

前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間におけるレジスタのデータ依存関係およびリソース競合に基づき、実行中の長語命令内の演算命令全ての実行が完了していなくとも前記命令バッファ内の長語命令の前記命令スケジュール部への発行を制御する手段を備えることを特徴とする情報処理装置。

【請求項2】 演算の種類毎に分類された n フィールド(n は1以上)の演算フィールドを有する長語命令からなる命令流 m 本(m は1以上)の処理を行なう情報処理装置であって、
 m 個の命令デコード部と、該命令デコード部から発行された演算命令を受ける前記演算の種類毎に設けられた n 個の命令スケジュール部と、該 n 個の命令スケジュール部対応にそれぞれ設けられ、対応する命令スケジュール

部から発行された演算命令を実行する実行部と、レジスタ部を備え、

前記各命令デコード部は、長語命令を格納する命令バッファと、実行中の長語命令と前記命令バッファ内の長語命令との間における、レジスタのデータ依存関係を判定する判定手段とリソース競合を判定する判定手段と、該両判定手段の判定結果がレジスタのデータ依存関係がなく且つリソース競合がないことを示すとき、実行中の長語命令内の演算命令全ての実行が完了していなくとも命令バッファ内の長語命令を各演算命令に分け対応する命令スケジュール部へ発行制御する手段を備え、

前記各命令スケジュール部は、前記命令デコード部から発行された演算命令を保持する命令スケジュールバッファと、命令スケジュールバッファに保持された演算命令の中から実行部に投入する演算命令を選択し、該選択した演算命令を命

令流番号と共に実行部に向け投入制御する手段を備え、
前記各実行部は、前記命令スケジュール部から投入される演算命令を実行する演算命令実行手段と、命令スケジュール部から投入される命令流番号を受け取りこれを保持する手段と、前記投入された演算命令と命令流番号から、前記演算命令実行手段の動作を制御する手段と、前記演算命令実行手段の状態を前記命令デコード部および命令スケジュール部に通知する手段を備えることを特徴とする情報処理装置。

【請求項3】 請求項2記載の情報処理装置において、
前記実行部の演算命令実行手段は、演算命令の実行により例外発生を検知したとき例外発生信号を前記命令流番号に対応する命令デコード部に通知し、

前記デコード部は、例外発生信号を通知した実行部に対して、前記全ての実行部の演算命令実行手段の状態が受け入れ可能状態のとき、例外発生をした演算命令の再実行を指示する手段を備えることを特徴とする情報処理装置。

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.